

Particle Swarm Algorithm for Improved Handling of the Mirrored Traveling Tournament Problem

Kevin TOLE*, Muhammed MILANI, Fullgence MWAKONDO

Abstract: In this study, we used a particle swarm optimization (PSO) algorithm to address a variation of the non-deterministic polynomial-time NP-hard traveling tournament problem, which determines the optimal schedule for a double round-robin tournament, for an even number of teams, to minimize the number of trips taken. Our proposed algorithm iteratively explored the search space with a swarm of particles to find near-optimal solutions. We also developed three techniques for updating the particle velocity to move towards optimal points, which randomly select and replace row and column parameters to find candidate positions close to an optimal solution. To further optimize the solution, we calculated the particle cost function, an important consideration within the problem conditions, for team revenues, fans, and media. We compared our computation results with two well-known meta-Heuristics: a genetics algorithm utilizing a swapping method and a Greedy Randomized Adaptive Search Procedure Iterated Local Search algorithm heuristic on a set of 20 teams. Ultimately, the PSO algorithm generated solutions that were comparable, and often superior, to the existing well-known solutions. Our results indicate that our proposed algorithm could aid in reducing the overall budget expenditures of international sports league organizations, which could enable significant monetary savings and increase profit margins.

Keywords: double-round robin tournament; home-away assignment; NP-hard; particle swarm optimization; traveling tournament problem

1 INTRODUCTION

Traveling tournament problems (TTPs) constitute an important class of non-deterministic polynomial-time (NP)-hard problems that have been thoroughly studied [1-4]. Generally, TTPs evaluate, and work to reduce, total team travel in league sports tournaments [5, 6], which have become a dominant global economic interest. Statistics provided by the Global Sports Market [7, 8] in 2018 showed a massive increase in global market revenue, from 46.5 to 90.9 billion USD in 2005 and 2017, respectively, causing a subsequent boost in the sports tourism industry [9]. In sports management, travel budgets are extremely costly, second only to payroll [10]. Thus, it benefits all teams, viewers, fans, leagues, sponsors, and media outlets to minimize travel and the associated costs, which can be done by optimizing game scheduling, especially for tournaments played in multiple locations [11]. Tournament success relies on well-scheduled games, which motivates researchers to find feasible near-optimal solutions for these combinatorial optimization problems.

A variety of tournament formats exist, including single round robin tournaments [12], double round robin tournaments (DRRTs) [13], single elimination tournaments (SETs) [14], and double elimination tournaments [15]. This study focuses primarily on DRRTs, as their NP-Hard nature presents a uniquely motivating challenge that has yet to be fully solved, despite most sports association leagues organizing such tournaments [13, 16].

Generally, a DRRT consists of teams that play amongst themselves. The games are usually scheduled as "home" or "away", which indicate games where a team hosts their opponent or plays at the opponent venue, respectively. Sports tournaments also typically include a few breaks, which are defined as three consecutive home or away games. A typical DRRT period is divided equally into halves, with all the first-half games mirrored in the second half so that each pair of teams plays twice during the tournament, once at each venue [17]. The main objective of a TTP, given these constraints, is to find an appropriate home-away assignment that minimizes the

total number of trips and/or the total distance travelled for all teams.

The TTP was initially introduced by Easton et al. [18] for scheduling Major League Baseball seasons, and they [19] subsequently proposed a hybrid branch-and-price algorithm, based on lower bounds that were set as the sum of minimum travel distances for each team. Another effective heuristic in sports scheduling was constraint programming, implemented by Henz et al. [20], who used activated propagator libraries to stabilize and perform the required search strategies. Significant improvements in computation time were later achieved by Nemhauser et al. [21]. In addition, Benoist et al. [22] utilized hybridized Lagrange relaxation and constraint programming that expanded to a global bound based on a hierarchical architecture, resulting in improved feasible solutions. Anagnostopoulos et al. [23] presented a simulated annealing heuristic that distinguished and categorized hard and soft constraints. They used a large neighborhood structure, with strategies such as oscillation and rehear, that balanced feasible and infeasible locations within the search space. Additionally, Ribeiro et al. [24] investigated mirrored TTP schedules. Recent advanced heuristic models, such as one developed by Tiago et al. [25], have utilized a novel local search heuristic, whose neighbourhood structure improves its reach within the solution space, compared to other models. Guillermo et al. [26] also designed a solution for the DRRT problem based on an integer programming model, which was successfully adopted by FIFA for its 2018 World Cup qualification process.

In our work, we propose a new TTP variant. Given a set number of teams in a mirrored DRRT, we are tasked with finding an appropriate home and away assignment schedule of play that minimizes the total number of trips taken. Notably, the TTP is a type of geometric graphical problem, similar to the well-studied traveling salesman problem (TSP) and vehicle routing problems, both of which require the total travel distance to be minimized [27] [28]. It can be considered surprising that solving the TTP is more difficult than the TSP [28] because it includes

multiple optimization constraints and general objectives regarding issues of fairness, logistics, economics, and organization.

Here, we first introduce the variant TTP and propose a particle swarm optimization (PSO) metaheuristic that searches for a near-optimal TTP solution. This stochastic optimization technique iteratively explores the search space, using a swarm of particles called a population of candidate solutions, according to certain update formulas. Specifically, at each step, individual particles move to their current best known region and swarm position. In this work, we designed three new techniques for updating the velocity of the particle as it moves towards the optimal point. We then further optimize the solution by calculating the particle cost function, an important consideration in the problem conditions. In our opinion, the results presented here may be impactful in reducing overall budget expenditures for organizations associated with international sports leagues, which could save millions of dollars and increase profit margins.

The remainder of this paper is organized as follows. Section 2 describes the detailed problem constraints, while Section 3 thoroughly explains our proposed algorithm. All experimental results are provided and discussed in Section 4. Finally, Section 5 concludes the paper and outlines future work.

2 PROBLEM FORMULATION

Let $T = \{1, 2, \dots, n\}$ be a set of n teams, where n is even. The DRRT structure requires that each team plays every other team both at home and away during the season. Because there are n teams, a total of $n(n-1)$ games are played by all teams. Notably, this structure is the same as a complete directed graph on n vertices. We set t as the current match day of the tournament, so $1 \leq t \leq 2n-2$. Let $D = [d_{ij}]$ be the symmetric matrix of distances between the home cities of all teams, where d_{ij} represents the distance between the home cities of team t_i and team t_j .

Over the course of the tournament, each team travels to away games and returns home before their next home game. Let $C_i(t)$ be the distance travelled by team i in slot t . To consider travel both before the first game and after the last game, the range of t is extended to include $t = 0$. Then, the distance travelled by team t_i in a complete tournament is:

$$K_i = \sum_{t=0}^{2n-3} C_i(t), \quad (1)$$

and the total distance travelled by all teams in a complete season is:

$$K = \sum_{i=1}^n \left(\sum_{t=0}^{2n-3} C_i(t) \right). \quad (2)$$

The TTP objective is minimizing K such that certain constraints are met by the teams. Herein, we formulate and discuss these constraints individually. Let $T_{i,t}$ be an indicator variable, such that $T_{i,t} = 1$ if team i plays at home

against away team $X_{i,t}$ during slot t , and $T_{i,t} = 0$ if team i plays an away game against team $X_{i,t}$ during slot t .

Constraint 1. No team can play against itself, so:

$$i \notin \{X \mid 0 \leq t \leq 2n-3\} \quad (3)$$

Constraint 2. Each team can only play once per time slot t . As a result, for each $i \in \{1, \dots, n\}$ and $t \in \{1, \dots, 2n-2\}$:

$$\sum_{i=1}^n P_{i,t} = 1, P_{i,t} = \begin{cases} 1 & \text{if } X_{i,t} = i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Constraint 3. Each team can play an opponent at home, or away, only once during the tournament. Therefore, only one game between team i and team j is played at home. From the perspective of the home team, this implies that for each $i, j \in \{1, \dots, n\}$:

$$\sum_{t=1}^{2n-2} P_{i,t} = 1, P_{i,t} = \begin{cases} T_{i,t} & \text{if } X_{i,t} = i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Constraint 4. An additional constraint, often used in literature and real-world tournaments alike, is that no team may play more than three consecutive home or away games. For $i \in \{1, \dots, n\}$ and $t \in \{1, \dots, 2n-6\}$:

$$1 \leq \sum_{s=t}^{t+4} T_{i,s} \leq 3. \quad (6)$$

These four constraints limit the solution space. To clarify the objective function, we now formulate functions $C_i(t)$, which reflect the distances teams travelled. We consider three situations: 1) travel before the first game is played, 2) travel after the last game is played, and 3) travel occurring between the first and last games. In addition, travel between games t and $t+1$ must also be considered. For modelling purposes, we require an additional boundary value, $t=0$, to include travel before the first game and after the last game. These endpoints are addressed as follows. For $t=1$, we consider team i . If the first game is played at home; then,

$$C_i(0) = 0 \quad (7)$$

If the first game is played away, however, the team must travel to the game; hence,

$$C_i(0) = d_{ip} \text{ where } p = X_{i,1} \quad (8)$$

Next, we suppose that $t = 2n-2$ and consider team i . If the last game is played at home; then,

$$C_i(2n-3) = 0 \quad (9)$$

Alternatively, if the last game is played away, the team must travel back home after the game; hence,

$$C_i(2n-3) = d_{ip} \text{ where } p = X_{i,2n-2} \quad (10)$$

For $t \in \{1, \dots, 2n - 4\}$, travel between games falls into four different categories, outlined here. A team with consecutive home games does not travel:

$$C_i(t) = 0, \text{ if } T_{i,t-1} = 1, T_{i,t} = 1 \quad (11)$$

a team with a home game followed by an away game travels to the game:

$$C_i(t) = d_{ip}, \text{ where } p = X_{i,t}, \text{ if } T_{i,t-1} = 1, \text{ and } T_{i,t} = 0 \quad (12)$$

a team with consecutive away games travels from its previous away opponent to the next:

$$C_i(t) = d_{pq}, \text{ where } p = X_{i,t-1}, q = X_{i,t}, \text{ if } T_{i,t-1} = 0, T_{i,t} = 0 \quad (13)$$

and finally, a team with an away game followed by a home game travels home:

$$C_i(t) = d_{pi}, \text{ where } p = X_{i,t-1}, \text{ if } T_{i,t-1} = 0, T_{i,t} = 1 \quad (14)$$

With the problem thus established, we can now develop and present an algorithm for minimizing K , in accordance with Eq. (2).

3 ALGORITHM

To develop our algorithm, we divided the TTP into two objectives: 1) to find an appropriate home-away assignment for team schedules and 2) to minimize the collective travel distance of all teams. Here, we introduce the definitions necessary for developing the algorithm.

Definition 1. Timetable. A timetable is a matrix with n rows and $2n - 2$ columns. Here, we assume that the set of rows is T , which contains values 1 to n , and the set of columns is W , which contains values 1 to $2n - 2$. Each entry in a timetable, (t, w) , where $(t, w) \in T \times W$, shows the opponent of team t in week w .

A timetable T should satisfy the following conditions: for each team $r \in T$, the r^{th} row of T is a permutation $\pi(\cdot)$ of $T \setminus \{r\}$; and for any $(r, s) \in T \times S$, $\pi(\pi(r, s), s) = r$.

Definition 2. Home-Away Assignment. A home-away assignment is a matrix $M = [m_{rs}]$ where $m_{rs} \in \{0, 1\}$, with rows indexed by T and columns indexed by S . If team r plays at home during slot s , then $m_{rs} = 1$. Otherwise, $m_{rs} = 0$. A detailed example is given in Tab. 1.

Table 1 Travelling table for four teams

	week 1	week 2	week 3	week 4	week 5	week 6
Team A	1	0	0	0	1	1
Team B	0	1	1	1	0	0
Team C	0	1	1	1	0	0
Team D	1	0	0	0	1	1

Definition 3. Schedule. A DRRT schedule is a timetable and home-away assignment pair. We introduce a traveling schedule timetable that constructs a team schedule timetable for this study, as shown in Tab. 2.

We also follow a heuristic rule, advised for the TTP, to have a consistent home-away assignment such that all teams have similar minimum total travel distances. If D_i is the total distance travelled by team i , then, given a

minimum $D_i = \sum_{t=1}^{2n-2} C_i(t)$, a chosen tolerance δ is expected

such that $|D_i - D_j| \leq \delta$ for any $j \neq i$. A sample possible traveling sequence table, for four teams, is illustrated in Tab. 1. The first column lists the team names, while columns 2 to 7 denote weeks, with slots of 0 and 1 representing home and away games, respectively. The tournament is split into two halves, with the first half mirrored in the second, as shown in the grey and green patterns in Tab. 2, respectively. Tab. 1 displays the corresponding match pairs, but they are inverted with respect to the home team.

Table 2 Tournament scheduling for four teams

	week 1	week 2	week 3	week 4	week 5	week 6
Team A	B	C	D	B	C	D
Team B	A	D	C	A	D	C
Team C	D	A	B	D	A	B
Team D	C	B	A	C	B	A

3.1 Particle Swarm Optimization

The PSO algorithm was initially designed by Kennedy and Eberhart [29, 30], who were inspired by the natural collective social behaviour of animal groups. This stochastic optimization metaheuristic involves a swarm of particles, initially distributed at random, moving iteratively through a search space. They explore the space based on update formulas, so that each particle moves from its current best-known region toward the best-known position within the entire solution space. Thus, by exploring the space, the particles iteratively converge to an optimal position. When needed, constraint-based PSOs limit the available space in the algorithm.

In our PSO implementation, each particle is influenced by the swarm velocity v . For each algorithm iteration, the particles move within the search space to find their global optimal position. Let $pBest$ be the current best position of a particle, based on its most recent iteration, and let $gBest$ be the best globally known solution among all particles. In each iteration, particles adjust their location by moving toward $gBest$ and redefining the candidate solution. In this way, a history of $pBest$ and $gBest$ locations is stored collectively by the particles, and the space is thus explored. As shown in the Algorithm 1 below, each calculation initiates a particle velocity update, which is discussed in detail in a subsequent section.

Algorithm 1 PSO Algorithm

```

PSO n, costValues#costValue is cost of traveling
for<n times>do
    Initialization          #See Algorithm 2
end
for<PSO repeat times>do
    for<each particle>do

         $v_i^{t+1} \rightarrow \text{update}(v_i^t)$  #Update the particle's velocity ( $v_i$ )
         $x_i \leftarrow x_i + v_i$       #Update the particle's position
                                #See Algorithm 3 for Updating process
        costposition  $\leftarrow$  Calculate the particle's cost
        if costposition is better than costpBest then
            pBest  $\leftarrow x_i$       #Update pBest
        if costposition is better than costgBest then
            gBest  $\leftarrow x_i$       #Update gBest
        end
    end
end
return gBest

```

3.2 Initialization

Here, we consider a table with N rows and M columns, corresponding to the number of teams and total games played by each team, respectively, so $M = 2 \cdot (N - 1)$. A table with random entries is referenced as the primitive "particle" in the algorithm. Initially, several of these particles were created using an $N \times M$ dynamic table and randomly assigning either 0 or 1 to each position, representing the home or away status of the game and the opponent. Using the mirroring property [31], only the first half of the table, rather than the entire table, needs to be generated. According to the problem definition, the three-dimensional table has a depth of $2N - 2$, with each depth representing the week in which the teams will play. As described in Section 2, the table has a total of $N \times N \times (2N - 2)$ cells, which can each be set to 0 or 1. To prepare this table, we define, for each i in the interval $1 \leq i \leq n$, a Team i set as weeks that have not yet been selected for i^{th} Team. Then, for each $Team_i$ and $Team_j$, we randomly select t as the week they should play together, so that $t \in Team_i \cap Team_j$, where t is selected randomly with a probability of $1/\|Team_i \cap Team_j\|$. We then randomly set the value of $X_{i,j,t}$ to 0 or 1 and $X_{j,i,t} = X'_{i,j,t}$. It should be noted that the above definition can be approached practically in another way. In this alternate definition, a matrix with n lines and $2N - 2$ columns is assumed to contain the rows and columns as the team and the week in which the teams will play, respectively. The $X_{i,t}$ value of the table indicates that the t^{th} Team will play at home or away in t^{th} week, based on a value of 0 or 1, respectively. From the definition of the mirroring problem, it is obvious that for $1 \leq i \leq n/2$ $X_{i+n/2,t} = X'_{i,t}$. In the following we also constructed a cost function that selects a particle cost value that approaches the global best at each iteration, assigned as $gBest$. Because our specific discrete problem is a minimization, the particle with the minimum cost is optimal by definition. Algorithm 2 provides the specific procedure for generating a particle and corresponding $pBest$ and setting $gBest$:

Algorithm 2 Particle initialization and $gBest$ setting

```

PERSON          #N: Number of Teams
Create a dynamic table of size N * M called position
for<x from 0 to N/2>do
  for<y from 0 to N-1>do
    value ← rand(0 or 1)
    if value is available to (x,y) in current table then
      position (x, y) ← value
    end
  else
    position (x, y) ← value'
  end
  position(x+N/2, y) ← position(x,y)
#Mirroring properties
position(x, y+N-1) ← position(x,y)
position(x+N/2, y+N-1) ← position(x,y)
end
end
for<x = 0 to N>do
  for<y = 0 to 2*N-2>do
    pBest(x, y) ← position(x, y)
  end
end
costposition ← calculate cost value of position
if costposition < costgBest then
  for<x = 0 to N>do
    for<y = 0 to 2*N-2>do
      gBest(x, y) ← position(x, y)
    end
  end
end
end

```

3.3 Particle Velocity Update

We calculated the velocity of all particles as they move toward their optimal points, with all initial velocities assumed to be zero. If the initial iteration is $i = 1$, then at the i^{th} iteration:

$$v_i^{t+1} = w \cdot v_i^t + c_1 r_1 (pBest_i^t - x_i^t) + c_2 r_2 (gBest_i^t - x_i^t) \quad (15)$$

Accordingly, particle velocities update based on the previous $pBest$ and $gBest$ values. In this study, we considered three sequential velocity updates:

- We updated the current position according to the $pBest$ particle. We selected a row and column randomly, as an r_1 parameter, and replaced the c_1 positions in the $pBest$ table from this location with the next value in the position table.

Algorithm 3 Particle position update algorithm

```

Obtain (x,p) randomly #Move in pBest direction
for<y = p to p+c>do
  value ← pBest(x,y)
  if value ≠ position(x,y) and position(x,y) available for value then
    position(x, y) ← value
    position(x+N/2, y) ← value' #Mirroring properties
    position(x, y+N-1) ← value'
    position(x+N/2, y+N-1) ← value
  end
end

Get (x,p) randomly #Move in gBest direction
for<y = p to p+c>do
  value ← gBest(x,y)
  if value ≠ position(x,y) and position(x,y) available for value then
    position(x, y) ← value
    position(x+N/2, y) ← value' #Mirroring properties
    position(x, y+N-1) ← value'
    position(x+N/2, y+N-1) ← value
  end
end

Get (x,y) randomly #Move in own direction
for<w times>do
  (posX,posY) ← Find suitable location posX and posY
  position(posX,posY) ← position(posX,posY)
  invert all mirroring of position(posX,posY)
end

```

- Then, we updated the current position according to the $gBest$ table. We again selected a row and column randomly, as an r_2 parameter, and then replaced the c_2 positions in the $gBest$ table from this location with the next value in the position table.
- Finally, we updated the current position according to the current table. We found the w of the current position that suits the change value and applied it.

Consequently, the parameters w , c_1 , and c_2 tune the algorithm to optimize its results. We use Algorithm 3 to calculate the new positions of current particles.

3.4 Particle Cost Calculation

Another integral component for PSO algorithms is cost. To create a cost function, we must consider the problem conditions. According to the problem definition, the traveling cost can be calculated as follows:

- For each away game a team travels to, the cost is 1, and this is constant for every trip that every team takes for an away game.
- When a team is at an opposing venue for an away game and then returns for a home game, the traveling cost is also 1.
- When a team plays an away game, and travels directly to a subsequent away game without returning home in between, the traveling cost is again 1.
- When a team plays a home game and is scheduled to host again in their next game, the traveling cost is null.
- During the final week of a tournament, if a team plays an away game and then returns home, the traveling cost is 1.

Algorithm 4 was used to calculate these travel costs.

Algorithm 4 Travel cost calculation algorithm

```

s ← 0 #counter variable
for<x from 0 to N>do
if position(x,0) = 1 then s ← s+1

for<y=0 to 2*N-2>do
if position(x,y-1) == 0 and position(x,y) == 1 then
s ← s+1
end
elseif position(x,y-1) == 1 and position(x,y) == 1 then
s ← s+1
end
elseif position(x,y-1) == 1 and position(x,y) == 0 then
s ← s+1
end
end
end

```

3.5 pBest and gBest Update

At the end of each particle movement, the *pBest* and *gBest* values are updated accordingly. To accomplish this, we decide the costs of the current particle, *pBest*, and *gBest*.

If the current cost value is lower than the *pBest* or *gBest* costs, the current cost and particle tables should be replaced by the *pBest* or *gBest* tables.

4 Computational Experiments

To evaluate our proposed approach, we implemented it for a TTP using C++ on a personal computer with 3.0 GHz of RAM and 4.0 GB of memory. We set parameters for tournaments with varying numbers of participating teams. For an example four-team tournament, we set the particle number to 4, w , c_1 and c_2 to 3, and the number of rounds to 80. We ran a program with these parameters ten times, to minimize any random effects. Tab. 3 summarizes the corresponding experimental setup.

Table 3 Experiments and computational results (w , c_1 and c_2 = PSO parameters)

n	Particle Number	w	c_1	c_2	Round	Worst Result	Best Result
4	4	3	3	3	80	17	16
6	4	5	5	5	120	48	48
8	4	7	7	7	160	80	80
10	8	9	9	9	200	126	120
12	8	11	11	11	240	193	192
14	8	13	13	13	280	260	252
16	20	15	15	15	320	334	321
18	20	17	17	17	360	436	432
20	40	19	19	19	400	525	520

We then used a set of 20 teams to compare our results with two previous benchmark studies, by Urrutia et al. [32] and by Tinnaluk et al. [33]. The computational results are summarized and compared directly in Table 4. The first column consists of n teams, followed directly by our original results (OR) in the second column. The third column presents the lower bounds (LB) results. The fourth and fifth columns provide the best-known results (BKR) from Tinnaluk et al. [33] and the known results (KR) from Urrutia et al. [32], respectively. Columns 6 to 8 indicate the relative gaps between the OR and LB, OR and KR, and OR and BKR, respectively. Lastly, column 9 denotes the number of Breaks (BS).

The results in Tab. 4 demonstrate that our proposed methods achieve significant improvements over the known results when n is 4, 10, 14, 16, 18, and 20 and over the best known results when n is 4, 10, 14, 16, and 20. In the few remaining cases, our methods generate the same solutions as the known benchmark values. Furthermore, the LB values confirm our solutions as optimal. We also note a large gap when there are 16 teams, which significantly decreases the total travel cost achieved by our method. Thus, in all cases, our proposed PSO algorithm performs as well as, or better than, traditional solution methods.

Table 4 Computational results (OR = our original results, LB = lower bounds, BKR = best-known results, BS = number of Breaks)

n	OR	LB	BKR	KR	Gap#1 (OR-LB)	Gap#2 (OR-KR)	Gap#3 (OR-BKR)	BS
4	16	16	17	17	0	-1	-1	8
6	48	48	48	48	0	0	0	11
8	80	80	80	80	0	0	0	31
10	120	120	130	130	0	-10	-10	49
12	192	192	192	192	0	0	0	66
14	252	252	253	256	0	-6	-1	110
16	321	320	348	342	+1	-21	-27	156
18	432	432	432	434	0	-2	0	176
20	520	520	521	526	0	-6	-1	227

Additionally, we can trace the decreasing cost value of the solution. Fig. 1 shows this decrease for the example of a 20-team tournament.

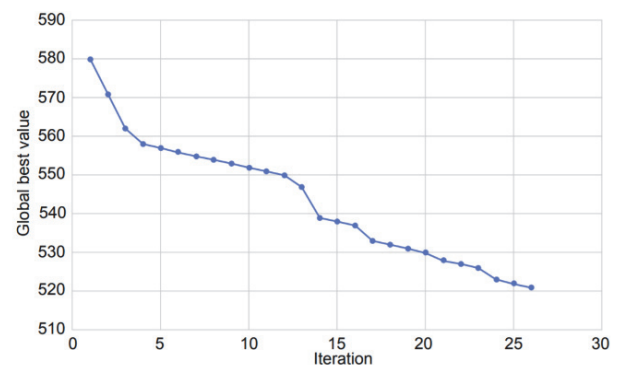


Figure 1 Decreasing cost for a 20-team tournament

5 CONCLUSION AND FUTURE WORK

Given its real-world applications, sports scheduling has become an important area within the international operations research community. In particular, TTPs have attracted significant attention because they are difficult to solve, despite their easy formulation. In this paper, we addressed a TTP that focuses on a mirrored DRRT. In this

case, the tournament is divided into two similar half-seasons, where corresponding match pairs are played but they are inverted with respect to the home team. We proposed a PSO metaheuristic for solving this combinatorial optimization problem, which searches for a home-away assignment that reduces the number of trips and subsequently minimizes travel expenditures. To facilitate the search space exploration, we designed three efficient strategies for updating the velocity of particles moving towards global minima. Furthermore, due to the constrained nature of the problem, we developed a cost function to further minimize the travel distance sums.

Computational experiments indicate that our results are comparable, and in some cases superior, to existing solutions for this problem. Notably, this problem remains a challenge to the field despite the significant improvements provided here, and further research is needed to generate high-quality solutions. Although we have proposed PSO metaheuristics here, we additionally plan to investigate other metaheuristics that not only apply to DRRT problems, but also augment and combine multiple tournament graphs within a single season of play. This problem poses a more significant challenge than the DRRT alone but has widespread potential applications and, therefore, warrants more attention. One classic example is the Union of European Football Associations Champions League, wherein the first round of play is a DRRT but the final stage converges to a SET. Finally, we believe that our manuscript makes significant contribution to the field of operation research and sports globally, as it provides quantifiable improvements over previous studies. Thus, our results may impact the overall expenditure budgets of sports league organizations, enabling monetary savings and improved profit margins by significantly reducing traveling costs.

6 REFERENCES

- [1] Easton, K., Nemhauser, G., & Trick, M. (2002, August). Solving the travelling tournament problem: A combined integer programming and constraint programming approach. *International Conference on the Practice and Theory of Automated Timetabling*, 100-109. https://doi.org/10.1007/978-3-540-45157-0_6
- [2] Khelifa, M., Boughaci, D., & Aïmeur, E. (2017, April). An enhanced genetic algorithm with a new crossover operator for the traveling tournament problem. *4th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1072-1077. <https://doi.org/10.1109/CoDIT.2017.8102741>
- [3] Goerigk, M. & Westphal, S. (2016). A combined local search and integer programming approach to the traveling tournament problem. *Annals of Operations Research*, 239(1), 343-354. <https://doi.org/10.1007/s10479-014-1586-6>
- [4] Durán, G., Durán, S., Marenco, J., Mascialino, F., & Rey, P. A. (2019). Scheduling Argentina's professional basketball leagues: A variation on the Travelling Tournament Problem. *European Journal of Operational Research*, 275(3), 1126-1138. <https://doi.org/10.1016/j.ejor.2018.12.018>
- [5] Bradbury, J. C. (2019). Determinants of revenue in sports leagues: An empirical assessment. *Economic Inquiry*, 57(1), 121-140. <https://doi.org/10.1111/ecin.12710>
- [6] Fang, D., Holmén, M., Kleinlercher, D., & Kirchler, M. (2017). How tournament incentives affect asset markets: A comparison between winner-take-all tournaments and elimination contests. *Journal of Economic Dynamics and Control*, 75, 1-27. <https://doi.org/10.1016/j.jedc.2016.11.006>
- [7] *Global sports market-total revenue from 2005 to 2017*. (2018). Statista. Retrieved from <https://www.statista.com/statistics/370560/worldwide-sports-market-revenue.html>
- [8] Kim, Y. H., Li, H., & Nauright, J. (2018). A destination development by building a brand image and sport event tourism: a case of Sport City USA. *Sport in Society*, 21(8), 1196-1203. <https://doi.org/10.1080/17430437.2018.1450718>
- [9] Yuan, J. & Liu, Y. (2019, March). Evaluation of Integration Development of Sports Industry and Tourism Industry: A Case Study of Guangdong Province. *5th International Conference on Information Management (ICIM)*, 346-351. <https://doi.org/10.1109/INFOMAN.2019.8714665>
- [10] Hu, C. & Cole, S. (2016). Comparing expenditure patterns and travel characteristics among NFL fans with different levels of team identification. *Tourism Review International*, 20(1), 3-10. <https://doi.org/10.3727/154427216X14581596798906>
- [11] Vellenga, D. B. (1971). Team travel - a major sports problem: Airline competition intense. *Business Horizons*, 14(6), 55-60. [https://doi.org/10.1016/0007-6813\(71\)90059-0](https://doi.org/10.1016/0007-6813(71)90059-0)
- [12] Suksompong, W. (2016). Scheduling asynchronous round-robin tournaments. *Operations Research Letters*, 44(1), 96-100. <https://doi.org/10.1016/j.orl.2015.12.008>
- [13] Carlsson, M., Johansson, M., & Larson, J. (2017). Scheduling double round-robin tournaments with divisional play using constraint programming. *European Journal of Operational Research*, 259(3), 1180-1190. <https://doi.org/10.1016/j.ejor.2016.11.033>
- [14] Stanton, I. & Williams, V. V. (2011). Manipulating single-elimination tournaments in the Braverman-Mossel model. *Workshop on Social Choice and Artificial Intelligence*, 87.
- [15] Hiew, B. Y., Tan, S. C., & Lim, W. S. (2017). A double-elimination-tournament-based competitive co-evolutionary artificial neural network classifier. *Neurocomputing*, 249, 345-356. <https://doi.org/10.1016/j.neucom.2016.11.082>
- [16] Rasmussen, R. V. & Trick, M. A. (2008). Round robin scheduling - a survey. *European Journal of Operational Research*, 188(3), 617-636. <https://doi.org/10.1016/j.ejor.2007.05.046>
- [17] Zeng, L. & Mizuno, S. (2013). On the generalized mirrored scheme for double round robin tournaments in sports scheduling. *Asia-Pacific Journal of Operational Research*, 30(03), 1340008. <https://doi.org/10.1142/S0217595913400083>
- [18] Easton, K., Nemhauser, G., & Trick, M. (2001, November). The traveling tournament problem description and benchmarks. *International Conference on Principles and Practice of Constraint Programming*, 580-584. https://doi.org/10.1007/3-540-45578-7_43
- [19] Easton, K., Nemhauser, G., & Trick, M. (2002). Solving the travelling tournament problem: A combined integer programming and constraint programming approach. *International Conference on the Practice and Theory of Automated Timetabling*, 100-109. https://doi.org/10.1007/978-3-540-45157-0_6
- [20] Henz, M. (2001). Scheduling a major college basketball conference revisited. *Operations research*, 49(1), 163-168. <https://doi.org/10.1287/opre.49.1.163.11193>
- [21] Nemhauser, G. L. & Trick, M. A. (1998). Scheduling a major college basketball conference. *Operations research*, 46(1), 1-8. <https://doi.org/10.1287/opre.46.1.1>
- [22] Benoist, T., Laburthe, F., & Rottembourg, B. (2001, April). Lagrange relaxation and constraint programming collaborative schemes for travelling tournament problems. *Proceedings CPAIOR*, 1, 15-26.
- [23] Anagnostopoulos, A., Michel, L., Van Hentenryck, P., & Vergados, Y. (2006). A simulated annealing approach to the traveling tournament problem. *Journal of Scheduling*, 9(2), 177-193. <https://doi.org/10.1007/s10951-006-7187-8>

- [24] Ribeiro, C. C. & Urrutia, S. (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179(3), 775-787.
<https://doi.org/10.1016/j.ejor.2005.03.061>
- [25] Januario, T. & Urrutia, S. (2016). A new neighborhood structure for round robin scheduling problems. *Computers & Operations Research*, 70, 127-139.
<https://doi.org/10.1016/j.cor.2015.12.016>
- [26] Durán, G., Guajardo, M., & Sauré, D. (2017). Scheduling the South American Qualifiers to the 2018 FIFA World Cup by integer programming. *European Journal of Operational Research*, 262(3), 1109-1115.
<https://doi.org/10.1016/j.ejor.2017.04.043>
- [27] Alves, R. M. & Lopes, C. R. (2015, May). Using genetic algorithms to minimize the distance and balance the routes for the multiple traveling salesman problem. *2015 IEEE Congress on Evolutionary Computation (CEC)*, 3171-3178.
<https://doi.org/10.1109/CEC.2015.7257285>
- [28] Akter, S., Nahar, N., ShahadatHossain, M., & Andersson, K. (2019, February). A New Crossover Technique to Improve Genetic Algorithm and Its Application to TSP. *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1-6.
<https://doi.org/10.1109/ECACE.2019.8679367>
- [29] Kennedy, J. & Eberhart, R. (1995, November). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942-1948.
- [30] Marini, F. & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149, 153-165.
<https://doi.org/10.1016/j.chemolab.2015.08.020>
- [31] Ribeiro, C. C. & Urrutia, S. (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, 179(3), 775-787.
<https://doi.org/10.1016/j.ejor.2005.03.061>
- [32] Urrutia, S. & Ribeiro, C. C. (2006). Maximizing breaks and bounding solutions to the mirrored traveling tournament problem. *Discrete Applied Mathematics*, 154(13), 1932-1938. <https://doi.org/10.1016/j.dam.2006.03.030>
- [33] Rutjanisarakul, T., & Jiarasuksakun, T. (2017). A Sport Tournament Scheduling by Genetic Algorithm with Swapping Method. <http://arxivst.com/papers/2017/04/17/a-sport-tournament-scheduling-by-genetic-algorithm-with-swapping-method/>

Contact information:

Kevin TOLE, PhD Candidate

(Corresponding author)

Technical University of Mombasa, Institute of Computing and Informatics,

P.O. Box 90420 - 80100, Mombasa Kenya

E-mail: ktole@tum.ac.ke

Muhammed MILANI, PhD, Assistant Professor

Bandırma Onyedi Eylül University, Faculty of Computer Engineering,

PK: 10200 Bandırma / BALIKESİR-Turkey

E-mail: mmilani@bandirma.edu.tr

Fullgence MWAKONDO, Dr. Lecture

Technical University of Mombasa, Institute of Computing and Informatics,

P.O. Box 90420 - 80100, Mombasa Kenya

E-mail: mwakondo@tum.ac.ke